Swimming Competition Tests Description

Jakub Szafarczyk

Konrad Wnuk

I. Team CRUD:

- postTeam

  Method: POST

  Endpoint: http://localhost:8080/teams

  Description:

  This request is used to create a new team in the system. It must contain all NotNull elements in request.

- getTeams

  Method: GET

  Endpoint: http://localhost:8080/teams

  Description:

  Retrieves a list of all teams currently stored in the system.

- getTeam6

  Method: GET

  Endpoint: http://localhost:8080/teams/6

  Description:

  Fetches detailed information about a specific team, in this case, the team with ID 6.

- putTeam6

  Method: PUT

  Endpoint: http://localhost:8080/teams/6

  Description:

  Updates the information for team with ID 6. The request body must contain all of the updated team data.

- deleteTeam6

  Method: DELETE

  Endpoint: http://localhost:8080/teams/6

  Description:

  Deletes the team with ID 6 from the system.

II. Coach CRUD:

- postCoach:

  Method: POST

  Endpoint: http://localhost:8080/coaches

  Description:

  Creates a new coach in the system. It must contain all NotNull elements in request.

- getCoaches:

  Method: GET

  Endpoint: http://localhost:8080/coaches

  Description:

  Retrieves a list of all coaches registered in the system.

- getCoach6

  Method: GET

  Endpoint: http://localhost:8080/coaches/6

  Description:

  Fetches detailed information for the coach with ID 6.

- putCoach6

  Method: PUT

  Endpoint: http://localhost:8080/coaches/6

  Description:

  Updates the information for coach with ID 6. The request body must contain all of the updated coach data.

- deleteCoach

  Method: DELETE

  Endpoint: http://localhost:8080/coaches/6

  Description:

  Deletes the coach with ID 6 from the system.

III. Competition CRUD:

- postCompetition

  Method: POST

  Endpoint: http://localhost:8080/competitions

  Description:

  Creates a new competition in the system. It must contain all NotNull elements in request.

- getCompetitions

  Method: GET

  Endpoint: http://localhost:8080/competitions

  Description:

  Retrieves a list of all competitions registered in the system.

- getCompetition6

  Method: GET

  Endpoint: http://localhost:8080/competitions/6

  Description:

  Fetches detailed information for the competition with ID 6.

- putCompetition6

  Method: PUT

  Endpoint: http://localhost:8080/competitions/6

  Description:

  Updates the information for competition with ID 6. The request body must contain all of the updated competition data.

- deleteCompetition6

  Method: DELETE

  Endpoint: http://localhost:8080/competitions/6

  Description:

  Deletes the competition with ID 6 from the system.

IV. Competitor CRUD:

- postCompetitor

  Method: POST

  Endpoint: http://localhost:8080/competitors

  Description:

  Creates a new competitor in the system. It must contain all NotNull elements in request.

- getCompetitors

  Method: GET

  Endpoint: http://localhost:8080/competitors

  Description:

  Retrieves a list of all competitors registered in the system.

- getCompetitor6

  Method: GET

  Endpoint: http://localhost:8080/competitors/6

  Description:

  Fetches detailed information for the competitor with ID 6.

- putCompetitor6

  Method: PUT

  Endpoint: http://localhost:8080/competitors/6

  Description:

  Updates the information for competitor with ID 6. The request body must contain all of the updated competitor data.

- deleteCompetitor6

  Method: DELETE

  Endpoint: http://localhost:8080/competitors/6

  Description:

  Deletes the competitor with ID 6 from the system.

V. Location CRUD:

- postLocation

  Method: POST

  Endpoint: http://localhost:8080/locations

  Description:

  Creates a new location in the system. It must contain all NotNull elements in request.

- getLocation

  Method: GET

  Endpoint: http://localhost:8080/locations

  Description:

  Retrieves a list of all locations registered in the system.

- getLoaction6

  Method: GET

  Endpoint: http://localhost:8080/locations/6

  Description:

  Fetches detailed information for the location with ID 6.

- putLocation6

  Method: PUT

  Endpoint: http://localhost:8080/locations/6

  Description:

  Updates the information for location with ID 6. The request body must contain all of the updated location data.

- deleteLocation6

  Method: DELETE

  Endpoint: http://localhost:8080/locations/6

  Description:

  Deletes the location with ID 6 from the system.

VI. Race CRUD:

- postRace

  Method: POST

  Endpoint: http://localhost:8080/races

  Description:

  Creates a new race in the system. It must contain all NotNull elements in request.

- getRaces

  Method: GET

  Endpoint: http://localhost:8080/races

  Description:

  Retrieves a list of all races registered in the system.

- getRace6

  Method: GET

  Endpoint: http://localhost:8080/races/6

  Description:

  Fetches detailed information for the race with ID 6.

- putRace6

  Method: PUT

  Endpoint: http://localhost:8080/races/6

  Description:

  Updates the information for race with ID 6. The request body must contain all of the updated race data.

- deleteRace6

  Method: DELETE

  Endpoint: http://localhost:8080/races/6

  Description:

  Deletes the race with ID 6 from the system.

VII. Result CRUD:

- postResult

  Method: POST

  Endpoint: http://localhost:8080/results

  Description:

  Creates a new result in the system. It must contain all NotNull elements in request.

- getResults

  Method: GET

  Endpoint: http://localhost:8080/results

  Description:

  Retrieves a list of all results registered in the system.

- getResult11

  Method: GET

  Endpoint: http://localhost:8080/results/11

  Description:

  Fetches detailed information for the result with ID 11.

- putResult11

  Method: PUT

  Endpoint: http://localhost:8080/results/11

  Description:

  Updates the information for result with ID 11. The request body must contain all of the updated result data.

- deleteResult11

  Method: DELETE

  Endpoint: http://localhost:8080/results/11

  Description:

  Deletes the result with ID 11 from the system.

VIII. Parameter Requests:

- team1Coaches

  Method: GET

  Endpoint: http://localhost:8080/teams/1/coaches

  Description:

  Retrieves a list of coaches assigned to team with ID 1.
  This request fetches elements (coaches) that are associated with another element (team).

- coach2Team

  Method: GET

  Endpoint: http://localhost:8080/coaches/2/team

  Description:

  Fetches the team assigned to coach with ID 2.
  This allows navigation from a coach to entity (team).

- competition3Races

  Method: GET

  Endpoint: http://localhost:8080/competitions/3/races

  Description:

  Returns all races that belong to competition with ID 3.
  Enables you to retrieve a nested list of races within a specific competition.

- competitor2Results

  Method: GET

  Endpoint: http://localhost:8080/competitors/2/results

  Description:

  Gets all results associated with competitor ID 2.
  This fetches related performance data linked to a competitor.

- location1Competitions

  Method: GET

  Endpoint: http://localhost:8080/locations/1/competitions

Description:

Retrieves all competitions that are hosted at location ID 1.
This is used to track which events are scheduled or have occurred at a specific venue.

- race2Results

  Method: GET

  Endpoint: http://localhost:8080/races/2/results

  Description:

Gets the results for race with ID 2.
This allows accessing performance outcomes for a particular race.

- result3Competitor

  Method: GET

  Endpoint: http://localhost:8080/results/3/competitor

  Description:

Retrieves the competitor associated with result ID 3.
This enables you to trace a result back to the athlete who achieved it.

IX. Exception Handling:

- Error404

  Method: DELETE

  Endpoint: http://localhost:8080/teams/77777

  Description:

Attempts to delete a non-existent team using an invalid ID (77777).
Expected to return HTTP 404 Not Found, validating that the server correctly handles attempts to access or delete resources that don't exist.

- Error400

  Method: POST

  Endpoint: http://localhost:8080/competitions

  Description:

Sends a malformed or incomplete request body when creating a competition (missing required fields like name).
Expected to return HTTP 400 Bad Request, confirming that the API validates input data and rejects bad requests appropriately.

- Error500

  Method: GET

  Endpoint: http://localhost:8080

  Description:

Triggers a server-side failure by sending an invalid payload with a GET request to the root endpoint.
Expected to return HTTP 500 Internal Server Error, testing how the system handles unexpected internal exceptions or logic failures.